# AppleScript ヘルプ

基本

## AppleScript について

Mac の使用中に、いくつかの作業を繰り返し実行していることに気付くことがあります。AppleScript を使用して、繰り返し実行する作業を自動化するスクリプトとアプリケーションを記述できます。

AppleScript は、OS X に組み込まれている、強力で多目的のスクリプト言語です。AppleScript を使ってショートカットを作成したり、繰り返しのタスクを自動化したりできます。また、カスタムアプリケーションを作成して、時間を大幅に節約することもできます。AppleScript では、プログラミング可能なアプリケーションを制御できます。

AppleScript のスクリプトは、オブジェクトに対してアクションを実行する、英語に似たコマンドで構成されます。アクションには、open、close、get the name of などがあります。オブジェクトには、application、window、preference setting などがあります。

1 つのスクリプトで 1 つまたは複数のアプリケーションを制御できます。アプリケーションがスクリプティング対応であるためには、そのアプリケーションに対してスクリプトで使用できる用語が定義されている必要があります。OS X には、いくつかのスクリプティング対応アプリケーションが付属しています。

また、スクリプトはネットワークを介してアプリケーションを制御できるため、複数のコンピュータで操作を自動化することもできます。

実際、ユーザは知らないうちに AppleScript を使用しています。システムとアプリケーションの機能の多くは、実際にはアプリケーション内部から呼び出される AppleScript スクリプトです。たとえば、このヘルプシステムでは、アプリケーションやファイルを開くリンクをクリックするときは必ず、そのアクションを実行するのは AppleScript スクリプトです。

**関連トピック**

AppleScript の動作
AppleScript エディタについて
AppleScript 言語について
インターネット上の AppleScript リソース
AppleScript Studio 開発環境

基本

## AppleScript エディタについて

「AppleScript エディタ」は、AppleScript のスクリプトを読み、書き、記録し、そして保存できる使いやすいスクリプティングアプリケーションです。

「AppleScript エディタ」の機能の一部を紹介します：

- 統合された、使いやすいインターフェイス
- スクリプト内のテキストを検索および置換できる
- 長いスクリプト文を自動的に折り返す

- 「取り消す」を複数回実行できる
- スクリプティングにかかる時間を短縮するショートカットメニュー
- アプリケーションサービスに対応
- ライブラリパレットを使ってアプリケーションスクリプトにアクセスできる
- 文の完成を支援するスクリプトアシスタント
- スクリプトの実行状況を確認できる「結果の履歴」と「イベントログの履歴」

**関連トピック**

AppleScript 言語について
AppleScript Studio 開発環境

---

基本

## AppleScript スクリプトを作成する

この AppleScript を活用するために、プログラミングについて理解しなければならないことは何もありません。事実、スクリプトを 1 語も記述する必要はありません。OS X には、すぐに使用できる数百の便利な AppleScript スクリプトが組み込まれており、さらに数千の AppleScript スクリプトを Web から入手できます。

もう少し詳しく知りたい場合は、既存のスクリプトを入手して、少しの変更でそれらをニーズに合わせることができます。

もちろん、独自のスクリプトを作成することによって、その利点を最大限に活かすこともできます。スクリプトの記述方法を習得すれば、作業の生産性をより高める便利なツールや、時間を節約する自動操作を作成することができます。

「AppleScript エディタ」を使って独自のスクリプトを変更または作成できます。「AppleScript エディタ」は、スクリプトを記述するシンプルな環境を備えています。「AppleScript ヘルプ」にある例はすべて、「AppleScript エディタ」を使用しています。

---

- 「AppleScript エディタ」を開きます。

  「AppleScript エディタ」ウインドウに、以下のテキストを入力します。

  ```
  say "Welcome to AppleScript!"
  ```

  「実行」をクリックします。

---

**関連トピック**

AppleScript エディタについて
AppleScript Studio 開発環境

---

基本

## AppleScript の動作

AppleScript は、「アップルイベント」と呼ばれるメッセージを送信することによって動作します。スクリプトを実行すると、AppleScript が、コマンドを実行するシステムまたはアプリケーションにアップルイベントを送信します。

多くの場合、アプリケーションやシステムは、AppleScript に結果を返します。たとえば、スクリプトがアプリケーションにオブジェクトの名前の取得要求を送信すると、アプリケーションがその名前を返します。コマンドを実行できない場合は、アプリケーションまたは AppleScript がエラーメッセージを返す場合があります。「AppleScript エディタ」に返される結果を表示することができます。

通常は、アップルイベントの仕組みを把握する必要はありません。AppleScript は、そのすべてをバックグラウンドで実行します。必要なことは、スクリプトを記述して実行することだけです。

**関連トピック**

AppleScript について
AppleScript 言語について

---

基本

## スクリプトアプリケーションについて

アプリケーションとして保存されるスクリプトを、「AppleScript アプリケーション」または「アプレット」と呼びます。アプレットは、開いたときに自動的に実行され、スクリプト内のコマンドを実行します。アプレットは、1 組のコマンドのみを含む小さい関数の場合も、数千行を含む複雑なアプリケーションの 場合もあります。

アプレットは、編集可能または実行専用として保存できます。編集可能アプレットにはスクリプトのテキストもそのまま含まれるため、いつでも編集したり再コンパイルしたりできます。アプレットを編集または変更するときは、「AppleScript エディタ」アイコンの上にドラッグします。スクリプトを実行専用として保存すると、編集可能なテキスト部分が削除され、ファイルの編集や再コンパイルはできなくなります。

編集可能として保存したアプレットを編集するには、そのアプレットを「AppleScript エディタ」または別の AppleScript 編集アプリケーションで開きます。アプレットは、そのアプレットを開いたときにスクリプトの実行を開始する、「on run」ハンドラを必要とします。

- 以下の例は、「on run」ハンドラを示しています。

```
on run
        --first command
        --second command
end run
```

**関連トピック**

ドロップレットについて
AppleScript ファイルタイプ

---

基本

## コンパイル済みスクリプトについて

コンパイル済みスクリプトは、「AppleScript エディタ」またはその他の AppleScript 編集アプリケーションで読み取ったり編集したりできる、編集可能な形式の AppleScript ファイルです。

コンパイル済みスクリプトを開いても、「AppleScript エディタ」には表示されますが、自動的には何のコマンドも実行しません。これにより、実際に実行せずに、スクリプトを変更することができます。次に、ファイル内の AppleScript コード

にすべてのテキストを読み取って変換するスクリプトをコンパイルします。コンパイル済みスクリプトを実行できます。コンパイル済みスクリプトのファイル拡張子は .scpt です。

ファイルをコンパイル済みスクリプトとして保存すると、そのテキストはコードと共にそのままファイル内に残ります。このため、コンパイル済みスクリプトは、いつでも再編集することができます。

**関連トピック**

スクリプトアプリケーションについて
ドロップレットについて
AppleScript ファイルタイプ

---

基本

## ドロップレットについて

ドロップレットは、そのアイコン上にファイルまたはフォルダをドラッグ＆ドロップしたときに実行される AppleScript スクリプトです。このスクリプトが、ファイルまたはフォルダ上でアクションを実行します。

アプレットと同様、ドロップレットは編集可能または実行専用として保存できます。編集可能ドロップレットにはスクリプトのテキストもそのまま含まれるため、いつでも編集したり再コンパイルしたりできます。ドロップレットを編集または変更するときは、「スクリプトエディタ」アイコンの上にドラッグします。スクリプトを実行専用として保存すると、編集可能なテキスト部分が削除され、ファイルの編集や再コンパイルはできなくなります。

ドロップレットは、そのドロップレットを開いたときにスクリプトの実行を開始する、「on open」ハンドラを必要とします。「on open」ハンドラには、ドロップレットにドラッグした項目のリストを含むパラメータがあります。ドロップレットスクリプトの一部はリストを調べ、別の部分が項目に影響を与えます。単に単一の項目に影響を与えるよう設計されているのでない限り、ドロップレットには、各項目のアクションを順番に実行してリスト内の項目全体にわたって繰り返す、繰り返しループが取り込まれるはずです。

- 以下に、「on open」とパラメータリストを含むドロップレットハンドラの例を示します。

```
on open (theItemsDropped)
        --first command
        --second command
end open
```

**関連トピック**

スクリプトアプリケーションについて
コンパイル済みスクリプトについて
AppleScript ファイルタイプ

---

基本

## AppleScript 用語説明を使用する

アプリケーションを制御するためのコマンドおよびオブジェクトを調べるには、そのアプリケーションの用語説明を参照します。用語説明には、アプリケーションが理解できる用語のセットが含まれます。

「ライブラリ」ウインドウでアプリケーションを選択するか、アプリケーションのアイコンを「AppleScript エディタ」ア

イコンにドラッグすることで、アプリケーションの用語説明を「AppleScript エディタ」に表示できます。

用語説明に含まれる用語は、「セット」とよばれる関連する用語のグループに分類されます。コマンドおよびオブジェクトは、ボールドテキストで表示されます。

---

1. 「AppleScript エディタ」で、「ウインドウ」＞「ライブラリ」と選択します。
2. 「ライブラリ」ウインドウで、アプリケーションの名前をダブルクリックします。
3. セットを選択します。

   セット内の用語およびその説明が表示されます。

---

AppleScript 言語は、次に示す異なる複数のカテゴリに分類されます。

- セット：関連する用語のグループ。AppleScript エディタの用語説明では、セットは最初のカラムに表示され、オレンジ色の「S」という文字で示されます。
- コマンド：アクションまたは結果を要求します。コマンドは「イベント」とも呼ばれ、2 番目のカラムに青色の「C」という文字で示されます。
- クラス：スクリプトに対応したオブジェクト。クラスの説明には、クラス内に存在する属性と要素がリスト表示されます。クラスは 2 番目のカラムに表示され、ラベンダー色の「C」という文字で示されます。
- 属性：値を含むオブジェクト。属性は 3 番目のカラムに表示され、ラベンダー色の「P」という文字で示されます。
- 要素：ほかのオブジェクトに含まれるオブジェクト。たとえば、段落には多数の単語を含めることができます。要素は、3 番目のカラムに表示され、オレンジ色の「E」という文字で示されます。

**関連トピック**

クラス定義
コマンド定義
AppleScript をコマンドアプリケーションに使用する

**開く**

「AppleScript Language Guide」：Commands Reference
「AppleScript Language Guide」：Class Reference

---

**基本**

# 問題解決

AppleScript を使うときに問題が起きる場合は、次のトピックを調べて問題を解決するための情報を探してください。

リストから選択するダイアログでキャンセルをクリックするとエラーメッセージが表示される場合

フォルダアクションが予期せずトリガされる場合

スクリプトがディスクを認識しない場合

---

**基本コマンド**

# クラス定義

クラス定義ではスクリプティング可能なオブジェクトの属性と要素を記述します。クラスとは文の中の名詞のようなもので、クラス定義とは辞書に書かれた名詞の定義のようなものです。

多くの場合、AppleScript 用語説明のクラス定義はボールドの用語で始まり、同じ行にその用語についての簡潔な説明が続きます。

次には、そのスクリプティング可能なオブジェクトに属する、またはそのオブジェクトを構成する、すべての要素のリストが記述されます。クラス定義が常に要素のリストを含むとは限りません。これは、スクリプティング可能なオブジェクトの中にはほかのスクリプティング可能なオブジェクトを含むものと含まないものがあるためです。

クラス定義の最後に記述されるのが、スクリプティング可能なオブジェクトのすべての属性と特性、およびそれらの値のリストです。

属性の各項目の先頭には属性の名前が付き、その後ろに以下の情報が続きます：

- 属性の値を保存するデータフォーマットの種類。整数（数）、ポイント（2 つの数の組み合わせ）、文字列（テキストデータ）、論理値（真または偽）など
- オプションのタイプインジケータ（[r/o]）。属性の値が、「読み出しのみ」（読むだけで変更できない）であるかどうかを表します
- 属性についての簡潔な記述

オブジェクトが複数形の場合は、複数のオブジェクトを参照できます。

要素のリストを持つオブジェクトには、ほかのオブジェクトが含まれます。スクリプトの中で使用できる要素の名前はボールドで表示されます。標準スタイルのテキストは要素を参照する方法を記述します。

オブジェクトには属性が定義されている場合があります。属性名はボールドで表示されます。標準スタイルのテキストは、文字列など、属性に含まれる値の種類を記述します。属性には、定数値または定数値のリストが含まれることがあります。属性の構文の末尾に「[r/o]」（読み出しのみ）という表示がなければ、スクリプトを使用して、その属性の値を取得したり、その属性の値を設定したりすることができます。

**関連トピック**

コマンド定義
オブジェクト定義
AppleScript コマンド
AppleScript 用語説明を使用する

**開く**

「AppleScript Language Guide」：Class Reference

---



# コマンド定義

コマンドは、アクションを実行する AppleScript 内の用語です。AppleScript 用語説明内のコマンドが、中央のカラムに表示され、青色の「C」という文字が付けられます。コマンドの定義には、コマンドの実行内容およびその構文の簡単な説明が含まれます。構文とは、コマンドをスクリプトに書き込むときの決まりのことです。構文は複数の部分から構成されていることがあります。

ボールドの単語は文の中で使用する用語です。用語説明に記載されている通りに正確に入力する必要があります。

青色のテキストの単語は、ユーザが指定する情報を示します。たとえば、ユーザが定義したオブジェクトや変数を識別します。

コマンドには省略可能な部分が含まれていることがあり、これらはかっこ内に表示されます。これらの部分は必ずしも文で使用する必要はありません。

**関連トピック**

クラス定義
AppleScript 用語説明を使用する
イベント定義
オブジェクト定義

**開く**

「AppleScript Language Guide」：Commands Reference

---



## イベント定義

イベントの定義は、スクリプティング可能なオブジェクトにアクションを適用するための動詞やコマンドを記述します。イベントとは、オブジェクトが実行、または参加できるアクションまたは動詞です。

多くの場合、イベントの定義は用語で始まり、続いて同じ行でその用語が簡潔に記述されます。

次に記述されるのは、そのコマンドの実行に使われたすべてのパラメータのリストです。オプションのパラメータはかっこで囲まれ、必須のパラメータは囲まれません。

それぞれのパラメータ項目はパラメータの名前で始まり、次がそのパラメータに使用されたデータフォーマットの種類で、パラメータの簡単な説明で終わります。

コマンドの定義には、そのコマンドが行うことと、その構文の簡単な説明（下線付きのテキスト）が含まれます。構文とは、コマンドをスクリプトに書き込むときの決まりのことです。構文は複数の部分から構成されていることがあります。

ボールドの単語は文の中で使用する用語です。用語説明に記載されている通りに正確に入力する必要があります。

標準スタイルのテキストの単語はユーザが指定する情報を示します。たとえば、ユーザが定義したオブジェクトや変数を識別します。

コマンドには省略可能な部分が含まれていることがあり、これらはかっこ内に表示されます。これらの部分は必ずしも文で使用する必要はありません。

コマンドはオブジェクトに対して実行されます。オブジェクトの定義には、名前や簡単な説明（下線付きのテキスト）など、オブジェクトを識別するために必要な構文が示されます。オブジェクトのクラスは、そのオブジェクトで使用できるコマンドなど、そのオブジェクトの属性を決定します。

**関連トピック**

データベースイベント
クラス定義
コマンド定義
オブジェクト定義
AppleScript の動作

# オブジェクト定義

コマンドはオブジェクトに対して実行されます。AppleScript 用語説明のオブジェクトの定義には、名前や簡単な説明（下線付きのテキスト）など、オブジェクトを識別するために必要な構文が示されます。オブジェクトのクラスは、そのオブジェクトで使用できるコマンドなど、そのオブジェクトの属性を決定します。

オブジェクトが複数形の場合は、複数のオブジェクトを参照できます。

要素のリストを持つオブジェクトには、ほかのオブジェクトが含まれます。スクリプトの中で使用できる要素の名前はボールドで表示されます。標準スタイルのテキストは要素を参照する方法を記述します。

オブジェクトには属性が定義されている場合があります。属性名は青色のボールドで表示されます。標準スタイルの青色のテキストは、文字列など、属性に含まれる値の種類を記述します。属性には、定数値または定数値のリストが含まれることがあります。属性の構文の末尾に「[r/o]」（読み込み専用）という表示がなければ、スクリプトを使用して、その属性の値を取得したり、その属性の値を設定したりすることができます。

**関連トピック**

AppleScript をコマンドアプリケーションに使用する
コマンド定義
クラス定義
オブジェクトの動作を指定する
AppleScript 用語説明を使用する

**開く**

「AppleScript Language Guide」：Class Reference
「AppleScript Language Guide」：Properties



# if 文

if 文は、特定の条件が満たされる場合にのみ実行されるコマンドです。それぞれの if 文には、true または false を評価するブール式が含まれます。if 文の内部のコマンドは、ブール式の値が true である場合にだけ実行されます。if 文は、end if で終わります。たとえば、次のように指定します：

```
if amountReceived > 100 then
        set memberStatus to "special"
end if
```

上の文では、ブール式が true である（amountReceived が 100 より大きい）場合に、if 文内部のコマンドが実行されます。ブール式が false の場合、コマンドは実行されません。

if 文に、else 文を組み合わせることもできます。この組み合わせは、if-then-else 文と呼ばれます。たとえば、次のような構文を使用します：

```
if amountReceived > 100 then
        set memberStatus to "special"
else
        set memberStatus to "general"
```

```
end if
```

上の例では、最初のブール式が true である（amountReceived が 100 より大きい）場合に、文の先頭部分のコマンドが実行されます。最初のブール式が false の場合、else に続く 2 番目の部分のコマンドが実行されます。

if 文を入れ子にすると、複数の条件を処理することができます。

AppleScript の if 文について詳しくは、「AppleScript Language Guide」を参照してください。

**関連トピック**

repeat 文
コマンド定義
AppleScript の変数を設定する
定数値

**開く**

「AppleScript Language Guide」 - If

---

基本コマンド

## repeat 文

repeat 文を使用すると、条件に応じて、タスクを複数回実行できます。repeat 文は、条件が満たされてループを抜けるまで、ひとまとまりの文の実行を続けます。repeat 文を入れ子にすると、さまざまな条件に対応することができます。

AppleScript には各種の repeat 文があり、さまざまな条件に対応できます。

### repeat (forever)

永久的に無限ループを繰り返します。ループを終了するには、exit 文を使用します。

```
repeat
        -- statement
        -- statement
        if (x = 5) then exit repeat
end repeat
```

### repeat (number) times

指定した回数だけ繰り返します。

```
repeat 5 times
        -- statement
        -- statement
end repeat
```

### repeat while

指定した条件が true の間、繰り返します。

```
set x to 1
```

```
repeat while (x < 5)
	-- statement
	-- statement
	set x to (x + 1)
end repeat
```

**repeat until**

指定した条件が true になるまで繰り返します。

```
set x to 1
repeat until (x = 5)
	-- statement
	-- statement
	set x to (x + 1)
end repeat
```

**repeat with (variable) from (startValue) to (stopValue)**

startValue に変数を設定してから、ループを通るたびに、その後でその変数を増分します。変数が stopValue に達すると、繰り返しは終了します。

```
repeat with x from (5) to (25)
	-- statement
	-- statement
end repeat
```

**repeat with (variable) in (list)**

リスト内の最初の項目に変数を設定します。その各パスは変数が次の項目に設定されているループを通ります。リストのすべての項目が使用されると、繰り返しは終了します。

```
set myList to {234, 342, 681, 874}
repeat with x in myList
	-- statement
	-- statement
end repeat
```

AppleScript の repeat 文について詳しくは、「AppleScript Language Guide」を参照してください。

**関連トピック**

if 文
AppleScript の変数を設定する
定数値

**開く**

「AppleScript Language Guide」 - repeat

# 定数値

定数は AppleScript の予約語で、既定値が含まれます。AppleScript には多数の定数が存在します。これらの定数を使って、オブジェクトの情報を取得したり、比較や算術計算を行ったりすることができます。アプリケーションおよびその他のオブジェクトでは、独自の定数を定義できます。これらの定数は、値クラス定義内で定義されます。

定数の値を調べるには、次の例を「AppleScript エディタ」にペーストして、「実行」をクリックします。「結果」をクリックして定数の値を表示します。

結果も定数です。これは、コマンドにより返された値を参照するスクリプト定数です。

```
get the current date
```

次に、定数のその他の例を示します：

- 算術定数：pi
- ブール定数：true、false
- 日付および時刻定数：minutes、hours、days、weeks
- 文字列定数：return、space、tab
- スクリプト定数：anything、current application、it、me、missing value、my、result

AppleScript の定数について詳しくは、「AppleScript Language Guide」を参照してください。

**関連トピック**

AppleScript の変数を設定する
オブジェクト定義
クラス定義
イベント定義

**開く**

「AppleScript Language Guide」 - constants



# AppleScript の変数を設定する

変数は、スクリプト内部で作成されるコンテナで、値の格納に使用されます。変数の作成時に、その名前を指定します。これは、識別子とも呼ばれます。スクリプトの実行時に、変数に値を設定したり、値にアクセスしたり、値を変更したりできます。変数は、スクリプトが実行されている間だけ存続します。スクリプトが閉じられると、値は失われます。

- 以下の例は、固有の識別子を指定して、それに値を割り当てる方法を示しています。

```
set myVar to "153"


set x to "Quarterly Reports"
```

```
set x to false
```

どのように値が変数に格納されるかを調べるには、これらの例を「AppleScript エディタ」にペーストして、「実行」をクリックします。「結果」をクリックすると、変数に格納された値が表示されます。

AppleScript の変数について詳しくは、「AppleScript Language Guide」を参照してください。

**関連トピック**

定数値
if 文
repeat 文

**開く**

「AppleScript Language Guide」 - variables

---

基本コマンド

## アプリケーションの場所を指定する

OS X のアプリケーションは「パッケージ」としてバンドルされています。パッケージとは、アプリケーションを構成するフォルダおよびファイルを含む特別なディレクトリのことです。このため、「path to」コマンドおよび「info for」コマンドは、アプリケーションを対象にして実行すると、異なる結果が返されることがあります。

アプリケーションを対象に「path to」コマンドを使用する場合、通常はパスの末尾が「.app」になります。Finder にはファイル拡張子がファイル名の一部として表示されないことがありますが、スクリプト内でファイルを識別するために拡張子を使用できます。

「info for」コマンドを使用して、アプリケーションについての情報を取得する場合、通常は、結果の「Folder」属性が「true」になります。また、「extension hidden」、「name extension」、または「displayed name」などの結果の追加属性を使用して、ファイルとその名前に関する詳しい情報を取得できます。

**関連トピック**

path to
Info for
フォルダ、ファイル、およびディスクを参照する
choose application
Finder のスクリプティングコマンド

---

Finder コマンド

## フォルダ、ファイル、およびディスクを参照する

AppleScript を使って、特定のフォルダ、ファイル、およびディスクを、パスを含めずに名前で参照することができます。AppleScript は、「書類」フォルダ、デスクトップ、ホームフォルダ、起動ディスクなどの項目の場所を識別できます。

- 以下の例は、スクリプトで名前を使用して、特定のフォルダを参照する方法を示しています。

```
set x to the path to the desktop folder
```

**関連トピック**

スクリプトで Finder を参照する
path to
Info for
Finder のスクリプティングコマンド

---



## スクリプトで Finder を参照する

Finder コマンドを使用するには、コマンドをアプリケーションに誘導するのと同様に、tell 文でコマンドを Finder に誘導します。

- 以下に、スクリプトでの tell 文の使用例を示します。tell 文は、ウインドウ数の機能を「Finder.app」に誘導します。

```
tell application "Finder.app"
        return the number of windows
end tell
```

**関連トピック**

Finder
オブジェクトの動作を指定する
フォルダ、ファイル、およびディスクを参照する
オブジェクト定義
Finder のスクリプティングコマンド

**開く**

「AppleScript Language Guide」 - Object specifiers

---



## Finder のスクリプティングコマンド

以下に、Finder で使用できるコマンドの一部を示します。

activate

alias file

alias list

application file

clean up

close

column

Computer-object

container

copy

count

data size

delete

desktop window

desktop

disk

document file

duplicate

eject

empty trash

exists

file

file information

find

Finder

Finder window

folder

get

icon view options

information window

Internet location file

item

list view options

make

move

moving folder window for

open

package

path to

preferences

preferences window

print

quit

フォルダ、ファイル、およびディスクを参照する

スクリプトで Finder を参照する

reveal

select

set

sort

trash

update

window

---

Finder コマンド

## 標準スクリプティング機能追加コマンド

スクリプティング機能追加によって、スクリプトに含めることのできる多くの便利なコマンドが追加されます。

adding folder items to

beep

choose application

choose color

choose file

choose file name

choose folder

choose from list

choose URL

clipboard info

close access

closing folder window for

current date

delay

dialog reply

display

display dialog

do shell script

download

FTP item

get eof

Info for

list disks

list folder

load script

mount volume

offset

open for access

open location

opening folder

path to

POSIX path

random number

read

round

run script

say

scripting components

set eof

set the clipboard to

set volume

store script

summarize

system attribute

time to GMT

upload

URL

write